BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problems Mailbox.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 978 981 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
09.02.2000 Bulletin 2000/06

(51) Int. Cl.[7]: **H04M 3/40**, H04M 3/42

(21) Application number: **99114250.6**

(22) Date of filing: **28.07.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **05.08.1998 US 129492**

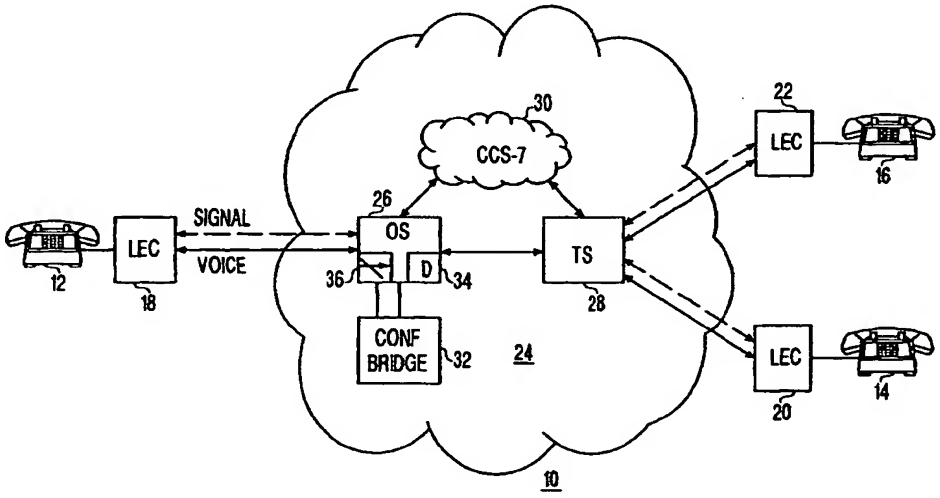
(71) Applicant: **AT&T Corp.**
New York, NY 10013-2412 (US)

(72) Inventor: **Eslambolchi, Hossein**
New Jersey 07920 (US)

(74) Representative:
Modiano, Guido, Dr.-Ing. et al
Modiano, Josif, Pisanty & Staub,
Baaderstrasse 3
80469 München (DE)

(54) Network-based caller speech muting

(57) A switch (26) in a telecommunications network (34) monitors a call placed by a calling party (12) to a called party (14) to detect the presence of prescribed condition, such as the entry by the caller of a self-muting signal, or an excessive level of background noise. Upon detecting the occurrence of the prescribed condition, the switch mutes the outgoing speech of the calling party while passing to the calling party the speech of the called party.



EP 0 978 981 A2

Description

Technical Field

[0001] This invention relates to a technique for muting a caller's outgoing speech.

Background Art

[0002] Many telephone callers must place calls from noisy environments. For example, travelers must often use pay phones or wireless terminals from locations such as airports or train stations, for example, that suffer from high ambient noise levels. The ambient noise at such locations often over powers the caller's speech, causing the telephone set to transmit noise rather than intelligible speech to the called party. The noise received from a caller originating a call from a noisy location is especially irritating during a conference call when the caller is but one of many participants.

[0003] Presently, various techniques exist to address the problem of ambient noise interfering with a caller's speech for calls originated from noisy locations. For example, AT&T now employs speech-processing equipment within its telecommunications network to filter callers' speech to reduce the effect of ambient noise. While such speech processing equipment is generally effective, high levels of ambient noise can defeat such filtering. Rather than utilize complex filtering techniques, subscribers who have telephone sets that offer a mute feature can self-mute their outgoing speech. Unfortunately, not all pay phones or wireless terminals offer muting capability so that a subscriber seeking to mute his or her speech must manually cover the terminal microphone, often at great inconvenience. Moreover, even if a calling party has a mute feature on his or her telephone, the caller may not know that the level of background noise is so excessive as to interfere with the caller's speech.

[0004] In connection with conference calls, U.S. Patent Application Serial No. 09/133118 (Attorney Docket No. Brown 1-6-10-38), filed on August 12, 1998, and assigned to AT&T, discloses a technique for enabling a first conference call participant to selectively mute other participants to the conference call. In this way, two or more participants can conduct a private conversation to the exclusion of all others. While the muting technique taught by Brown et al. affords the opportunity to mute individual callers, the technique does so only in connection with a conference call, rather than a conventional two-party call. Moreover, the muting technique of Brown et al. mutes both the incoming and outgoing speech of participants. Thus, Brown et al. provides no mechanism to allow a caller to self-mute only outgoing speech, nor does Brown et al. accomplish automatic muting of only outgoing speech in response to a high noise level.

[0005] Thus, there is need for a technique for enabling a caller to self-mute outgoing speech while still receiving speech from a called party.

Brief Summary of the Invention

[0006] Briefly, in accordance with the invention, a method is provided for muting a caller's outgoing speech. To mute a caller's outgoing speech, a telecommunication switch in a network monitors the call originated by the caller for a prescribed condition. The prescribed condition may comprise receipt of an in-band self-muting signal, typically in the form of a particular Dual Tone Multi-Frequency (DTMF) sequence, say #1 or *1 for example, or an out-of band self-muting signal, such as an Integrated Services Digital network (ISDN) call set up message. In response to the self-muting signal, the switch mutes the speech originating by the calling party while passing to the calling party the speech from each called party. In this way, the caller may self-mute his or her speech while continuing to hear speech from each called party.

[0007] The prescribed condition may also include excessive noise received from the caller, such as may occur when the caller calls from a noisy location. When the switch detects excessive noise, then the switch mutes the caller's outgoing speech even though the called party did not generate a self-mute signal. While the switch mutes the caller's outgoing speech, the caller continues to receive speech from each called party.

Brief Summary of the Drawing

[0008]

FIGURE 1 shows a block schematic diagram of a telecommunications network for practicing the present invention.

Detailed Description

[0009] FIGURE 1 shows a telecommunications network 10 for completing a call between a calling party, represented by telephone set 12, and one or more called parties, represented by telephone sets 14 and 16. Each of telephone sets 12, 14, and 16 may comprise a conventional analog station set, or an ISDN terminal. In the illustrated embodiment, the telephone sets 12, 14, and 16 receive local service (dial tone) from Local Exchange Carriers (LEC) 18, 20 and 22, respectively. It should be understood that any of the LECs 18, 20, and 22 could serve two or more of the telephone sets 14, 16, and 18.

[0010] An Inter-eXChange (IXC) network 24, such as the IXC network maintained by AT&T, typically carries calls between telephone sets served by different LECs, such as telephone sets 12 and 14 served by LECS 18 and 20, respectively. A call originated by telephone set 12 passes from the serving LEC 18 to the IXC network 24 for receipt in the network at an originating toll switch,

such as switch 26 that is associated or "homed" with the LEC 18. The originating switching switch 26 routes the call originated by the telephone set 12 to a terminating toll switch associated with the LEC serving the called party, such as the LEC 20 that serves the telephone set 14. In the illustrated embodiment, the originating toll switch 26 routes the call to the terminating toll switch 28 directly, although in practice, the originating switch may route the call through one or more intermediate (via) toll switches (not shown). Both the switches 24 and 26 typically comprise Model 4ESS telephone switches available from Lucent Technologies.

[0011] In addition to the toll switches 26 and 28, the IXC network 24 also includes a signaling network, such as AT&T's Common Channel Signaling (CCS) network. The signaling network 30 carries out-of band signaling messages to and from the switches 26 and 28 (as well as the via switches) to facilitate set-up and tear down of calls, as well as other logic associated with call flow.

[0012] The IXC network 24 may also include at least one conference bridge, such as the conference bridge 32 associated with the toll switch 26. The conference bridge 32 allows bridging of callers, thus permitting three or more parties on a single call. Although the illustrated embodiment of network 24 depicts a single conference bridge 32, in actuality, the network will typically include a plurality of bridges, each associated with a toll switch.

[0013] Not infrequently, the network 10 may carry a call originated at a telephone set, such as telephone set 12, situated in a location subject to high levels of ambient noise, such as an airport, train station, bus terminal, or along a busy street. The high level of background noise often interferes with the calling party's speech and may cause the network to transmit the noise, thus interfering with the called party's ability comprehend the called party's speech. A high level of noise from a calling party is especially irritating during a conference call because such noise may drown out the speech of other participants.

[0014] In accordance with the invention, the IXC network 24 advantageously overcomes the problem of high background noise levels interfering with a caller's speech by muting a caller's outgoing speech in response to a predetermined condition. As described below, the prescribed condition may comprise receipt of an in-band self-muting signal, typically in the form of a particular DTMF sequence, "*1" for example, or an out-of band signal, such as an ISDN call set up message. The prescribed condition may also include excessive noise received from the calling party as may occur when the party originates the call from a noisy location.

[0015] To monitor for the above-described conditions, each originating switch, such as OS 26, includes a detector 34 that monitors an incoming phone call to detect excessive levels of noise. The detector 34 also monitors the call for to detect a self-muting signal entered by the calling party. The calling party-entered self-muting signal may comprise an in-band signal, typically in the form of a particular Dual Tone Multi-Frequency (DTMF) sequence, say #1 for example, or an out-of band signal, such as an Integrated Services Digital network (ISDN) call set up message. (To the extent that switch 28 originates calls from one or both of the telephone sets 14 and 16, the switch would also include a detector having the same functionality as detector 34.)

[0016] In response to the excessive noise and/or receipt of a self muting signal, the detector 34 signals an outgoing path connection mechanism 36 that switches the caller's outgoing speech to mute such speech. The outgoing path connection mechanism 36 mutes the caller's outgoing speech until the detector 34 detects a drop in the background noise level below a prescribed threshold or until the detector detects a signal from the calling party to cancel the self-muting. The self-muting cancellation signal could comprise a particular DTMF sequence, such as #1, or an out-of-band ISDN signal. Upon detecting such a change in the prescribed condition, the detector 34 signals the outgoing path connection mechanism 36 to pass the caller's outgoing speech.

[0017] During intervals while the outgoing path connection mechanism 36 mutes the calling party's outgoing speech, the calling party receives incoming speech from the called party. Thus, regardless of a high background noise level, or the desire of the calling party to mute his or her speech, the calling party continues to her the called party. In many instances, a calling party desires to listen more than talk, especially in the course of a conference call. Thus, the combination of the detector 34 and outgoing path connection mechanism 36 allows a calling party participating in a conference call to listen, but not talk.

[0018] The foregoing provides a technique for muting the outgoing speech of a calling party in response to a prescribed condition while permitting the calling party to receive the speech of each called party.

[0019] Where technical features mentioned in any claim are followed by reference signs, those reference signs have been included for the sole purpose of increasing the intelligibility of the claims and accordingly, such reference signs do not have any limiting effect on the scope of each element identified by way or example by such reference signs.

Claims

1. In connection with a telephone call carried in a telecommunications network from a calling party to a called party, a method for muting the calling party's outgoing speech, comprising the steps of:

   monitoring at a switch in the network for a prescribed condition during the call; and

   responsive to the occurrence of said prescribed condition, muting speech from said calling party to said called party while passing

speech from said called party to said calling party.

2. The method according to claim 1 wherein the switch monitors for background noise that exceeds a prescribed level.

3. The method according to claim 1 wherein the switch monitors for entry by the calling party of a self-muting signal.

4. The method according to claim 3 wherein the switch monitors for entry of an in-band self-muting signal.

5. The method according to claim 4 wherein the switch monitors for entry of a prescribed sequence of Dual-Tone Multi-Frequency signals.

6. The method according to claim 3 wherein the switch monitors for entry of an out-of-band self-muting signal.

7. In connection with a conference call carried in a telecommunications network between a plurality of conference call participants, a method for muting at least one conference call participant's outgoing speech, comprising the steps of:

monitoring at a switch in the network for a prescribed condition during a call from said one participant; and
responsive to the occurrence of said prescribed condition, muting speech from said one participant to other conference call participants party while passing speech from said other conference call participants to said one conference call participant.

8. The method according to claim 7 wherein the switch monitors for background noise that exceeds a prescribed level.

9. The method according to claim 7 wherein the switch monitors for entry by said one conference call participant of a self-muting signal.

10. The method according to claim 9 wherein the switch monitors for entry of an in-band self-muting signal.

11. The method according to claim 10 wherein the switch monitors for entry of a prescribed sequence of Dual-Tone Multi-Frequency signals.

12. The method according to claim 9 wherein the switch monitors for entry of an out-of-band self-muting signal.

13. A. telecommunications switch for routing a telephone call from a calling party to a called party, including

means for monitoring for a prescribed condition occurring during the call; and
means responsive to said monitoring means for muting speech from calling party to said called party while passing speech from said called party to said calling party.

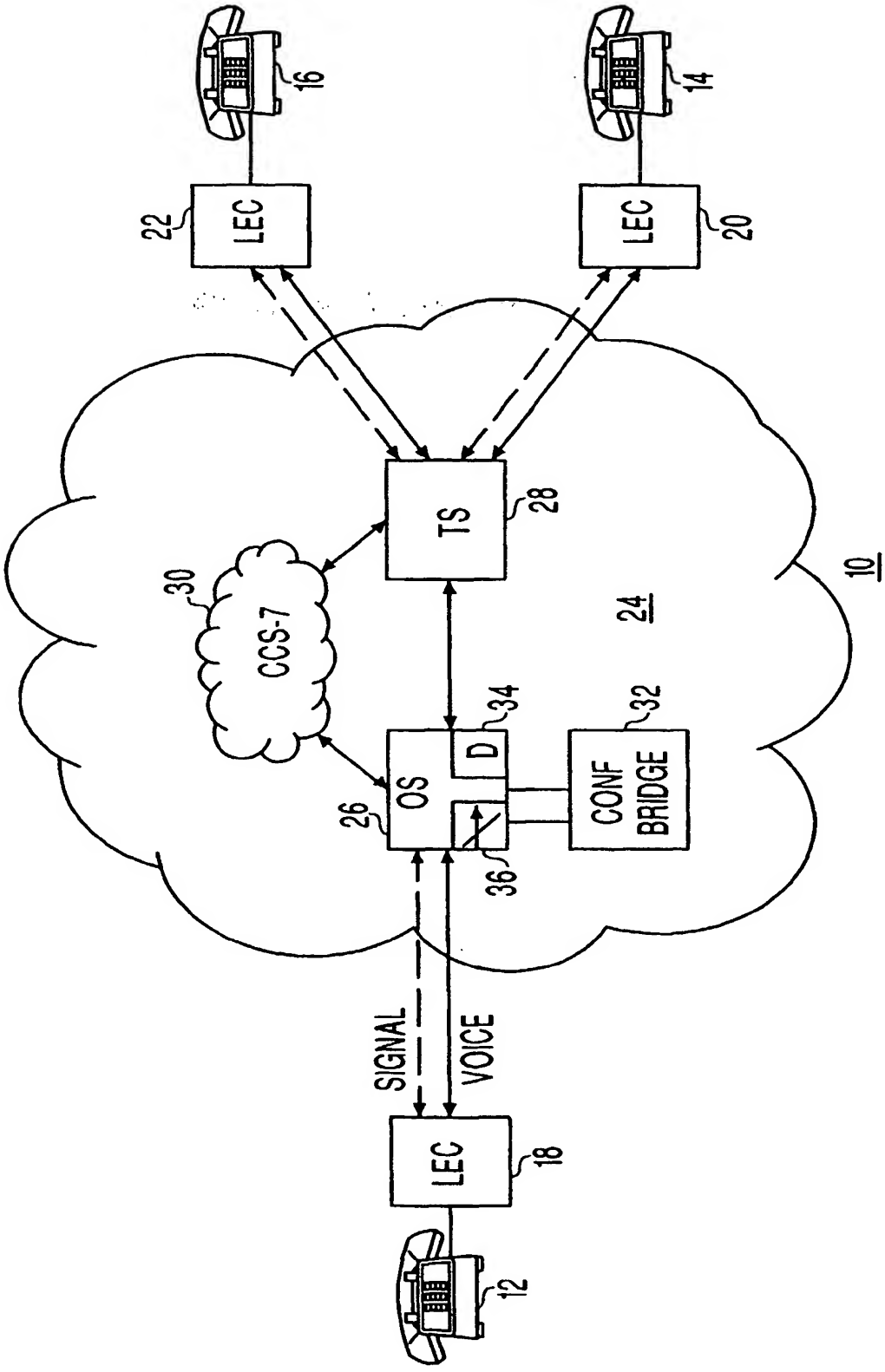
14. The apparatus according to claim 13 where said first means comprises a detector for determining if background noise originating from said calling party exceeds a prescribed level.

15. The apparatus according to claim 13 where said first means comprises a detector for detecting is said calling party has entered a self-mute signal.

16. The apparatus according to claim 15 wherein the detector detects the entry of an in-band self-muting signal.

17. The apparatus according to claim 16 wherein the detector detects the entry of a prescribed sequence of Dual-Tone Multi-Frequency signals.

18. The apparatus according to claim 15 wherein said detector detects the entry of an out-of band self-muting signal.



(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 664 636 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **94307408.8**

(51) Int. Cl.[6]: **H04M 3/56, H04L 12/18**

(22) Date of filing: **10.10.94**

(30) Priority: **19.01.94 GB 9400965**

(43) Date of publication of application:
26.07.95 Bulletin 95/30

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504 (US)

(72) Inventor: **Sharman, Richard Anthony**
18 Blenheim Avenue,
Highfield
Southampton,
Hampshire SO2 1DU (GB)
Inventor: **Adams, Paul Stuart**
32 Tees Close,
Chandlers Ford
Eastleigh,
Hampshire SO5 3RU (GB)

(74) Representative: **Davies, Simon Robert**
I B M
UK Intellectual Property Department
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) Audio conferencing system.

(57) A computer workstation supports speech recognition software 50 and conferencing software 45, and is involved in an audio conference with one or more other workstations. Speech from the user at that workstation is transmitted to the other workstation(s), and also converted into text by the speech recognition software. The conferencing software then transmits the text to the other workstation(s). Likewise, the conferencing software also receives the text equivalent of spoken contributions from the other workstation(s). This received text, together with the locally generated text, is stored in a text file so as to produce a set of minutes for the audio conference.

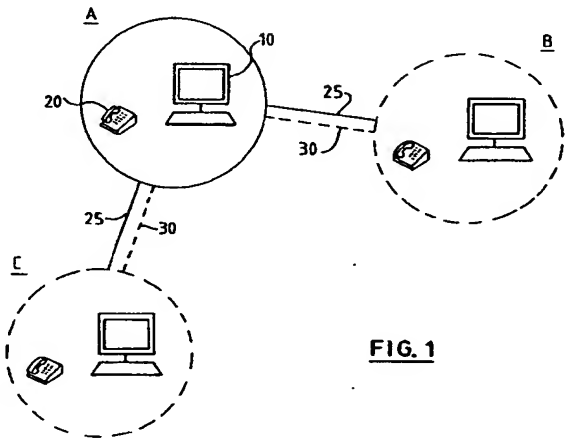


FIG. 1

EP 0 664 636 A2

The present invention relates to an audio conferencing system, and in particular to a method of textually recording at a workstation spoken contributions to an audio conference, each participant in the conference having an associated workstation, the workstations being linked together by one or more networks.

In recent years there has been a significant improvement in the performance of automatic speech recognition systems. Commercially available systems such as the Personal Dictation System (IPDS) from IBM are capable of recognising natural language, providing the words are spoken discretely (ie the words are not run together but rather there is a distinct gap between adjacent words). Future development will of course further enhance the capabilities of such systems over the coming years, for example to allow full recognition of continuous speech. Automatic speech recognition systems such as the above-mentioned IPDS from IBM are now being offered for use as dictation machines, whereby a person dictating a letter or other document speaks to the system, which then automatically converts the speech into text See "Computers that Listen", p30-35, New Scientist 4 December 93 for additional background information on such systems.

Whilst such a facility is clearly very powerful, there are some limitations on current technology that are not likely to be overcome in the foreseeable future. An example of such a restriction is for example where there are several speakers in a meeting, and to correctly minute the meeting there is a need to determine who is speaking at any particular time. In such circumstances a human recorder would typically rely on both visual and aural information in order to attribute speech to the correct speaker. Clearly an automatic speech recognition is unable to take advantage of such extra information, and so is unable to to replace a human recorder for this type of work.

Another area of technology which has seen considerable development over the past few years is teleconferencing. The driving force behind this activity is the recognition that face to face meetings, especially those which involve international journeys, are not only expensive, but also the excessive travelling necessarily wastes considerable time. It is therefore common nowadays for organisations to provide video teleconferencing suites, typically allowing parties in two or more remote sites to effectively hold a meeting together, despite their disparate locations.

The video suites required for conventional teleconferencing require expensive equipment and investment. Very recently therefore there has been a move to develop desk-top conferencing systems. Such systems exploit the fact that it is common for business people to have their own personal computer or workstation on their desk, and that these workstations are increasingly being linked together by various types of network, eg local area networks (LANs), or integrated services digital network (ISDN). The addition of suitable audio and video hardware to these workstations allows a distributed and highly flexible teleconferencing system to be provided. Examples of such multimedia conferencing systems are described in "Distributed Multiparty Desktop Conferencing System: MERMAID" by K Watabe, S Sakata, K Maeno, H Fukuoka, and T Ohmori, p27-38 in CSCW '90 (Proceedings of the Conference on Computer-Supported Cooperative Work, 1990, Los Angeles); "Personal Multimedia Multipoint Communications Services for Broadband Networks" by E Addeo, A Gelman and A Dayao, p53-57 in Vol 1, IEEE GLOBECOM, 1988; and "Personal Multimedia-Multipoint Teleconference System" by H Tanigawa, T Arikawa, S Masaki, and K Shimamura, p1127-1134 in IEEE INFOCOM 91, Proceedings Vol 3.
A distributed audio conferencing system is described in US 5127001.

JP-2-260750-A describes a conferencing system in which each terminal is fitted with a controller. The terminal with the loudest output is fed to a speech-to-text conversion unit, which is subsequently used to make a record of the conference. JP-2-260751 describes a conferencing system in which a speech buffer is provided for each terminal. The stored speech is then directed to a central speech-to-text unit when there is no voice activity at the associated terminal. Although these two applications teach a basic facility for minuting meetings, they suffer from a lack of flexibility and non-optimum usage of speech recognition systems.

Accordingly, the invention provides a method of textually recording at a workstation spoken contributions to an audio conference, each participant in the conference having an associated workstation, the workstations being linked together by one or more networks, the method comprising the steps of:

receiving local speech input at the workstation;

performing speech recognition on the local speech input at the workstation to generate a local text equivalent;

transmitting the local speech input to the other participant(s) in the conference;

receiving spoken contributions from the other participant(s) in the conference plus the corresponding text equivalents transmitted from the workstation associated with the respective participant;

storing both the local text equivalents and the text equivalents received from the other workstation(s) in a text file.

The audio conference itself can be implemented either over the network linking the workstations together, or over a separate network, for example using a conventional telephone conference. In the latter case the local speech input must be detected both by the conferencing system (eg telephone) and the microphone associated with the workstation for input into the speech recognition system. There is no requirement for any video conferencing facility, although this could be added to the system if desired.

The invention provides a distributed system for recording the minutes at a meeting, relying on real-time speech recognition at each participating node. Speech recognition has such a wide range of use that it will effectively become a standard feature of personal computers. By performing local speech recognition, the quality of the audio input signal is maximised (eg it is not distorted by transmission over the telephone network). Furthermore, each speech recognition systems can be trained to the user of that particular workstation; such speaker-dependent recognition offers improved accuracy over speaker-independent recognition. Another important aspect is that by using speech recognition in the desk top conferencing environment, the problem of attributing speech to different parties is readily solved; at any given workstation only the speech from the user of that workstation is converted into text, and this can then be readily marked with an indicator of origin (such as the name of the speaker or workstation). Thus when the text equivalents are combined into a single record in the text file, they already contain information identifying their source.

The only drawback of the local speech recognition is that the transmission of text format in addition to the audio could be regarded as redundant, although the extra bandwidth required by the text format is negligible. Conceivably one could drop the audio transmission, relying completely on the text format, which would be reconstituted into audio format at each receiving workstation using speech synthesis; however this is not very practicable, since the processing delay and recognition inaccurracies prevent any natural conversation, at least with current technology (moreover, future development of communications links is likely to provide ample bandwidth for audio transmissions). Nevertheless, such an approach may possibly be of interest for multilingual conferences, when an automatic translation unit could be interposed between the speech recognition and speech synthesis to convert the text into the correct language for each participant, although it will be appreciated that such a system is still some way off in the future.

Generally each text equivalent of a spoken contribution stored in said text file is accompanied by the time of the contribution and/or an indication of the origin of that spoken contribution, thereby providing an accurate record of the conference. Normally the indication of origin of a spoken contribution will be the name of the participant, but may also be the identity of the workstation from which the message was transmitted, if the former information is not available. The time recorded may be the time at which the message containing that contribution was received or alternatively, the time at which the text equivalent was actually generated at the originating workstation. The latter approach is more accurate, but requires the time to be included in the message itself. In general it will be necessary to edit the minutes text file after completion, for example to correct inaccuracies in the speech recognition. This can be performed jointly by all the participants in the conference using a shared editor to produce a single agreed set of minutes.

In a preferred embodiment the method further comprises the step of visually displaying at the workstation both the local text equivalents and the text equivalents received from the other workstation(s). This is useful if a participant in the conference has impaired hearing, or is having to understand a foreign language, in which case the displayed text may be easier to comprehend than the speech itself. Moreover, it provides a real-time indication to the participants of the text that is being recorded in the minutes.

In a preferred embodiment the text equivalents are visually displayed in a set of parallel columns, whereby each column displays the text equivalents of the spoken contributions from a single workstation. Preferably the method further includes the step of adjusting the cursor position within each of the columns after each new spoken contribution has been displayed to maintain horizontal alignment between the columns with regard to time synchronisation. Thus when read down the display the different contributions are correctly sequenced according to the order in which they were made.

Preferably the method further comprises the step of transmitting the local text equivalent of said local speech input to the other workstation(s) in the conference. This is useful for example to allow the other workstation(s) to display the text of spoken contributions made at the local workstation. The other workstation(s) could of course form their own set of minutes, although this might prove confusing and it may be best from a practical point of view to agree on just one node recording the minutes. To facilitate this the text recording process can be turned on and off during the audio conference (ie typically only a single node will turn on the text recording process). Note also that the ability to only record selected portions of the conference is

useful to prevent the minutes becoming excessively long. Typically text recording might be turned on after a point has been discussed to allow the conclusions and any necessary actions arising therefrom to be minuted.

The invention further provides a system for textually recording at a workstation spoken contributions to an audio conference, each participant in the conference having an associated workstation, the workstations being linked together by one or more networks, the method comprising the steps of:

means for receiving local speech input at the workstation;

means for performing speech recognition on the local speech input at the workstation to generate a local text equivalent;

means for transmitting the local speech input to the other participant(s) in the conference;

means for receiving spoken contributions from the other participant(s) in the conference plus the corresponding text equivalents transmitted from the workstation associated with the respective participant;

means for storing both the local text equivalents and the text equivalents received from the other workstation(s) in a text file.

An embodiment of the invention will now be described by way of example with reference to the following drawings:

Figure 1 is a schematic representation of an audio conference;

Figure 2 is a simplified block diagram of the major software components running on a node in the network of Figure 1;

Figure 3 is a simplified block diagram of a computer workstation for use in the conference of Figure 1;

Figures 4-8 are flow charts illustrating various aspects of the text recording application of Figure 2; and

Figure 9 illustrates the display presented to the user by the text recording application of Figure 2.

Figure 1 is a schematic representation of an audio conference between parties A, B, and C. Each party is equipped with a computer workstation 10 and a telephone 20. As explained in more detail below, the computer workstations are running the Person to Person (P2P) desktop conferencing program available from IBM. This provides for the exchange of messages (and optionally video) between the three parties. The messages are transmitted over links 25, which may form part of a local area network (LAN) or an integrated services digital network (ISDN), be asynchronous lines, or any other form of link supported by P2P. Note that there is no requirement under P2P for the conference to be over a homogeneous network; thus the link from A to B might be a LAN connection, whilst the link from A to C might be an ISDN connection. It will be appreciated that although Figure 1 shows only three parties, the invention is not so limited: P2P provides support for a conference of up to six parties, and even larger conferences may be available using other analogous software products.

The three parties A, B and C are participating in a three-way conference, whereby each workstation receives the audio signals of all the other workstations in the conference. The P2P software does not provide for voice communications (although this may change with future versions). Therefore, the audio part of the conference is implemented using standard telephones linked together using conventional telephone conferencing technology. The telephone connections are shown separately using dotted lines 30 in Figure 1. However, it would also be possible to transmit the audio signal over the same links 25 that connect the workstations together: for example, the transmission of audio over a LAN is described in "Using Local Area Networks for Carrying Online Voice" by D Cohen, pages 13-21 and "Voice Transmission over an Ethernet Backbone" by P Ravasio, R Marcogliese, and R Novarese, pages 39-65, both in "Local Computer Networks" (edited by P Ravasio, G Hopkins, and N Naffah; North Holland, 1982). Likewise, audio+data transmission over ISDN is the subject of relevant CCITT standards, whilst if the computer workstations are linked by asynchronous lines, modern modems such as WaveRunner from IBM are capable of multiplexing data and voice over the same link. Some commercially available conferencing software systems provide automatically for voice and data communications. Note also that the audio conferencing may be implemented either as a centralised system in which audio signals from each node go to a central multipoint control unit, where they are summed together before distribution to the participants in the conference, or as a distributed system in which each node broadcasts its audio signal direct to every other node in the conference. The particular architecture and format of the audio conference are not material to the present invention.

Figure 2 is a simplified block diagram of the main software components running on a workstation 10 participating in the conference of Figure 1. The operating system 45 is responsible for the basic functions of the computer in accordance with known techniques, and is also responsible for providing a graphical user interface whereby the user can perceive and enter information on a display screen. A suitable operating system is the multitasking OS/2 operating system available from IBM.

Above the operating system are three applications. The first of these is the speech recognition software 50, in this case the IBM Personal Dictation System (IPDS). The purpose of this software is to receive audio input and convert it into text in accordance with known speech recognition principles. The second application is the conferencing software 45, in this case the Person to Person software product, also available from IBM. Note that the only essential requirement of the conferencing system is that it is capable of sending text messages from one machine to another effectively in real-time. This facility is easily achieved using known computer communication techniques. The conferencing software may also provide audio and video communications, although the former can be provided using conventional telephones, as shown in Figure 1, whilst the latter is purely optional. The final application, the text recording application 40, effectively controls the speech recognition software and the conferencing software, and is described in more detail below. Also included in Figure 2 are device drivers 60, which allow the operating system as well as the speech recognition software and the conferencing software to interact with particular hardware components of the computer workstation in accordance with known techniques.

Figure 3 is a simplified schematic diagram of a computer system which may be used in the network of Figure 1. The computer has a system unit 110, a display screen 112, a keyboard 114 and a mouse 116. The system unit 110 includes microprocessor 122, semi-conductor memory (ROM/RAM) 124, and a bus over which data is transferred 126. The computer of Figure 3 is typically a workstation such as an IBM PS/2 Model 95 computer (note that the IPDS speech recognition software 50 requires a reasonably powerful workstation, typically having at least equivalent performance to the above-mentioned IBM PS/2 Model 95 computer). The computer of Figure 3 is equipped with two adapter cards. The first of these is a network adapter card 130, which together with accompanying software allows messages to be transmitted to and received from the other workstations shown in Figure 1. The network adapter card may be Token Ring (LAN) or ISDN, both available from IBM, or any other suitable communications device. The workstation may also have more than one adapter card to support a plurality of different communication formats. The operation of such network adapter cards is well-known and so will not be described in more detail. The second card is a speech recognition card 128 which is connected to a microphone for audio input. This card performs the digital sampling and encoding of the incoming audio, plus some of the processing associated with speech recognition. Assuming that the speech recognition software shown in Figure 2 is the above-mentioned IPDS, then the speech recognition card will be a Martin_LC card (also available from IBM).

The text recording application 40 will now be described in more detail with reference to Figures 4 to 8. This application is eventdriven; ie it responds to events or messages from other programs in accordance with known computer programming techniques. Figure 4 is a high-level diagram of the behaviour of the text recording application. Essentially the application first performs an initialisation (step 400), before holding the conference (step 410). During the conference minutes can be created; the minute recording function can be turned off and on during the conference. After the conference has concluded there is a chance to edit the minutes (step 420), prior to termination of the conference application (step 430).

Figure 5 shows the initialisation and termination procedures of the hold conference portion 410 of the text recording application. The conference starts with a general initialisation (step 510), followed by calls to start the P2P conferencing software 45 (step 520) and the IPDS speech recognition software 50 (step 530). Once this has been achieved, the system is now ready to receive events for processing (step 540), as described in more detail below. To terminate the conference the opposite actions are taken: ie the text recording software disconnects from the speech recognition software (step 550) and terminates the conference call (step 560).

The processing of locally generated events by the text recording application is shown in Figure 6A. Essentially events may be received either in response to user input from the keyboard of the workstation (610), or from the speech recognition software (620). The former occurs when the user wants to type something for display to the other participants in the conference (ie equivalent to a standard "talk" function). The latter arises from the user speaking as part of the audio conference. His or her speech is picked up by the microphone and passed to the speech recognition software for conversion into text. Each time the speech recognition software decodes another word, it raises an event 620, allowing the decoded word to be passed to the text recording application.

The text recording application therefore obtains text input 630, whether the result of speech recognition or typed user input. This text input is then formatted (step 650) and displayed in a window (step 660) corresponding to the local node (the display presented to the user is described in more detail below, but essentially there is one window for each participant in the conference, including the participant at the local node). Prior to this display, the cursor position in the local window is synch-

ronised with that of the remote windows (step 640), in other words, the cursor is moved to just beneath the bottom of any text in any of the other windows. This is achieved by maintaining a global variable indicating the position of the most recently written text in any window, which is updated each time new text is written to a window. Thus the cursor position can be synchronised with the other windows in accordance with the value stored in this global variable.

The text recording application also includes a timer, which regularly generates an event after a predetermined time interval. Whenever the text recording application receives an event from the timer 670 (see Figure 6B), it takes any text added to the local window since the last such event (step 680), and submits it to the conferencing software (step 690), which then forwards it to the other participants in the conference using known messaging techniques. In the current embodiment the predetermined time interval is 5 seconds. Note that if this value is shorter, more messages are sent over the network, adding to the bandwidth required; alternatively, if this value is increased, the delay between audio or text entry at the local node and display at a remote node gets longer which can result in usability problems. The selection of 5 seconds for the predetermined interval is therefore a compromise between these two factors, but can be tuned to individual circumstances. In particular, if bandwidth and other transmission overheads are not a problem, a shorter value could be used.

Figure 7 shows the receipt of a message from a remote node in the form of an event 700 received from the conferencing software. The actual transmission of messages between the local and remote nodes (and vice versa) is well-known in the art and will not be described in more detail. The conferencing software then passes the received message plus an identification of the originating node to the text recording application (step 710). The text recording application then synchronises the local and remote windows again (step 720): as described above, this essentially means that the cursor position is moved to immediately below the most recent addition to any window. The received text can then be formatted (step 730) and added to the window corresponding to the originating node (step 740).

The actions shown in Figure 7 lead to a textual display of the spoken contributions to the conference, but no permanent record. The latter is achieved by the process of Figure 8, which is invoked by turning on the create minutes facility (see step 410 in Figure 4). This then opens a file (step 810) in which to record the minutes, and writing suitable header information such as the time and date at the top of the file (step 820). The text written onto the screen is then read from the two windows using standard programming techniques (step 830), and copied to the minutes file together with the date, time, and identification of the node from which or participant from whom they originated (step 850). The point at which text recording starts is under user control (note that even text that has scrolled out of the window and is no longer visible on the actual display remains available). The process essentially scans all windows in parallel, ie one line at a time across all windows. Because the different contributions are synchronised on writing to the windows, and assuming only one person is speaking at any given time, the output from this scanning process will contain each of the different contributions, already correctly ordered. This process continues until the create minutes facility is turned off, the available text has all been recorded (step 840) or the conference concludes, leading to the closure of the minutes file (step 860).

It is then possible to edit the minutes file using a conventional text editor. This can be done whilst the conference is still in progress, eg the minutes are read out so that everyone can hear and agree to them. It is also possible to use a shared editor, in which the set of minutes is simultaneously displayed on all workstations and can be jointly edited. It is straightforward to implement such a shared editor using the P2P conferencing software.

Note that the time associated with each entry into the minutes is the arrival time of the message containing the entry at the local node (ie the node recording the minutes). Clearly due to transmission delays this will be slightly later than the actual time at which the relevant words were spoken. If necessary this could be overcome by including in each text message sent between nodes the time at which the message was originally generated. This time could then be entered into the minutes in association with the corresponding text, rather than a locally generated time, and the minutes text file then correctly sequenced in a subsequent processing stage. Note also that because text is accumulated in a 5 second buffer prior to transmission over the network, this delay may conceivably lead to some uncertainty about the relative timings of very brief remarks (eg person A may speak followed by person B, but the 5 second timer in B's machine expires first, leading eventually to the comment from B being entered into the minutes before that of A). If such problems do arise they can be easily rectified by reducing the period of the timer from 5 seconds to a smaller value.

Figure 9 illustrates a typical screen displayed at the workstation for a two-way conference, and in particular the window 900 for the text recording application. This window includes buttons 905 in the top left and right corners for performing sizing

and other operations on the window, in accordance with standard programming techniques. The top bar 910 of the window shows the name of the text recording application ("Minute Man"), plus the name that has been given to this particular conference ("Status check"). Beneath this is a menu bar 920 which if selected present the user with a set of possible actions (as is standard for computer programs). The different options available are:

File: this is used to set or change the name of the conference (currently "Status Check), control printing options, etc

Edit: this provides standard editing operations such as cut-and-paste which allow the text displayed to be manipulated

Voice: this is used to initiate the speech recognition software, identify the speaker to it, and so on

Windows: this option is standard with P2P applications, and allows the user to see which other applications are active

P2P: this option is used to control the conference, for example to add or remove parties from the conference

Minutes: this option is used to start generating an actual text file of minutes from the text displayed on the screen, and can also be used for example to invoke a shared editor to edit the minutes

Help: standard help function

Note that the lowest bar 970 in the window is used to provide one line of text information about an option whenever a particular option is selected from the menu bar 920 (ie it is a rudimentary form of help).

The next bar 930 in the window provides simple status information about the conference. First listed is the name of the local conference participant and node ("Paul" and "Lucretia" respectively). The "03" identifies which P2P conference this is (it is possible to run more than one conference simultaneously). Next the two participants are listed plus their machines, and finally it is indicated that this is the first instance of Minute Man running (again it is possible to run more than one in parallel).

Next two bars 940, 950 identify the participants and their machines (obviously the number of bars displayed here would correspond to the number of participants in the conference). Beneath each of the participant bars is a window 945, 955 containing the recognised speech of that participant. It will be noted how the contributions of each participant are spaced as described above, to provide proper sequencing of the different contributions. Between the two windows is a scroll bar 960, which can be used to scroll the contents of the two windows forwards or backwards, thereby permitting earlier text to be reviewed

The facility to turn the minute recording function on and off is useful to avoid excessive amounts of text being recorded. Typically it is most efficient to discuss a point, and then minute a summary of the conclusion of the discussion plus any associated actions. This can be carefully entered using discrete speech to ensure optimum speech recognition. This also minimises the need for reviewing large quantities of text generated by the speech recognition unit, which may well be rather inaccurate since the participants are unlikely to maintain careful speech all through the conference (note that if they in fact slip into continuous speech, the IPDS will simply ignore this as noise). Normally a single set of minutes will be created at one workstation in a conference, which can be edited and agreed upon by all the participants. This set of minutes is then distributed to the other workstations, which therefore do not need to generate their own separate set.

Although in the above-described embodiment the speech recognition software resides on the same physical workstation as the conferencing software, it is also contemplated that a client-server architecture may be used, whereby the speech recognition software is located on a separate server machine. Typically the server machine would have high processing speeds, and be connected to the client workstations by a high bandwidth LAN. It would then be fed the audio input, and return the text of the recognised speech to the workstation, effectively in real-time.

Claims

1. A method of textually recording at a workstation spoken contributions to an audio conference, each participant in the conference having an associated workstation, the workstations being linked together by one or more networks, the method comprising the steps of:

   receiving local speech input at the workstation;

   performing speech recognition on the local speech input at the workstation to generate a local text equivalent;

   transmitting the local speech input to the other participant(s) in the conference;

   receiving spoken contributions from the other participant(s) in the conference plus the corresponding text equivalents transmitted from the workstation associated with the respective participant;

   storing both the local text equivalents and the text equivalents received from the other workstation(s) in a text file.

2. The method of claim 1, wherein each text equivalent of a spoken contribution stored in said text file is accompanied by the time of the contribution and/or an indication of the origin of that spoken contribution.

3. The method of claim 1 or claim 2, further comprising the step of transmitting the local text equivalent of said local speech input to the other workstation(s) in the conference.

4. The method of claim 3 wherein said step of transmitting the local text equivalent to the other workstations occurs regularly after a predetermined time interval.

5. The method of any preceding claim, further comprising the step of visually displaying at the workstation both the local text equivalents and the text equivalents received from the other workstation(s).

6. The method of claim 5, wherein each text equivalent of a spoken contribution is displayed accompanied by the time of the contribution and/or an indication of the origin of that spoken contribution.

7. The method of claim 6, wherein the text equivalents are visually displayed in a set of parallel columns, whereby each column displays the text equivalents of the spoken contributions from a single workstation.

8. The method of claim 7, further including the step of adjusting the cursor position within each of the columns after each new spoken contribution has been displayed to maintain horizontal alignment between the columns with regard to time synchronisation.

9. The method of any preceding claim, further comprising the step of editing said text file.

10. The method of any preceding claim, wherein the text recording process can be turned on and off during the audio conference.

11. A system for textually recording at a workstation spoken contributions to an audio conference, each participant in the conference having an associated workstation, the workstations being linked together by one or more networks, the method comprising the steps of:

means for receiving local speech input at the workstation;

means for performing speech recognition on the local speech input at the workstation to generate a local text equivalent;

means for transmitting the local speech input to the other participant(s) in the conference;

means for receiving spoken contributions from the other participant(s) in the conference plus the corresponding text equivalents transmitted from the workstation associated with the respective participant;

means for storing both the local text equivalents and the text equivalents received from the other workstation(s) in a text file.

12. The system of claim 11, further comprising means for transmitting the local text equivalent of said local speech input to the other workstation(s) in the conference.

13. The system of claim 11 or claim 12, further comprising means for visually displaying at the workstation both the local text equivalents and the text equivalents received from the other workstation(s).

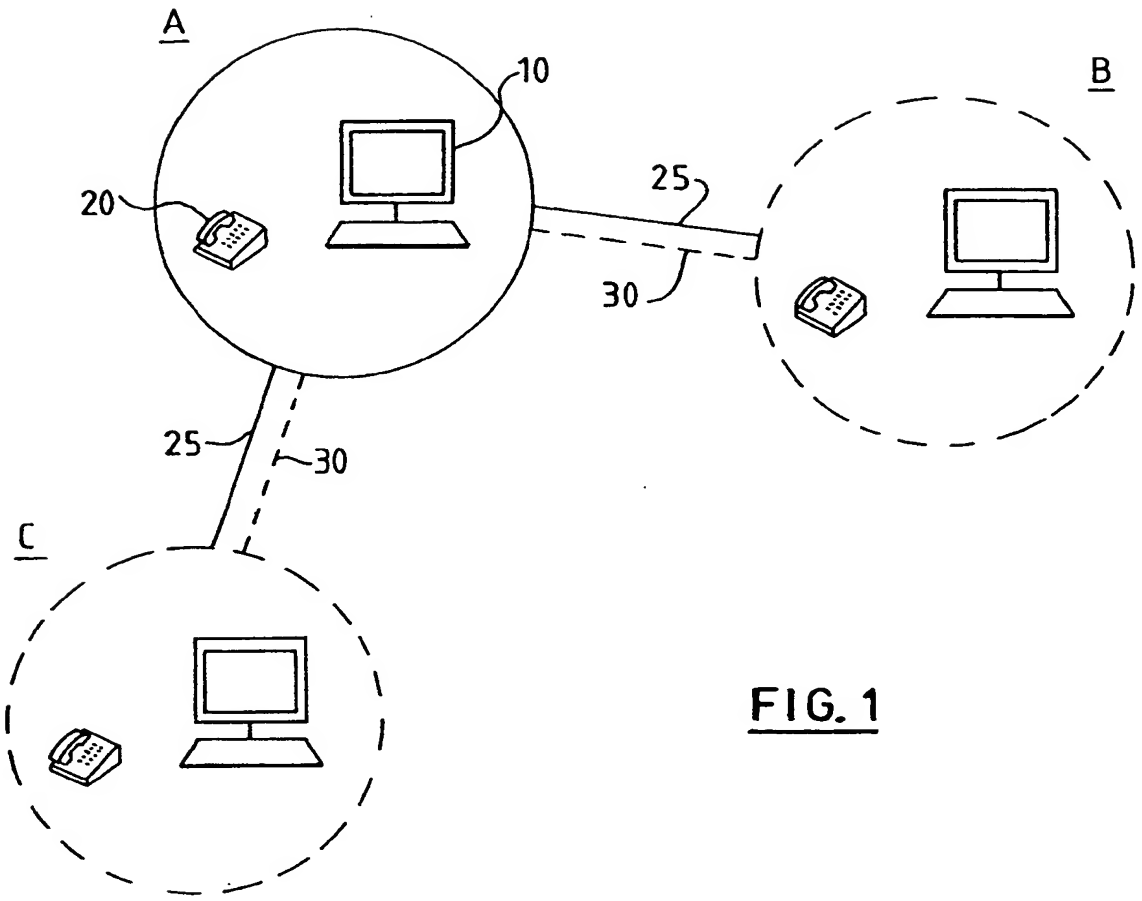

A
10
B
20
25
30
25
30
C

FIG. 1

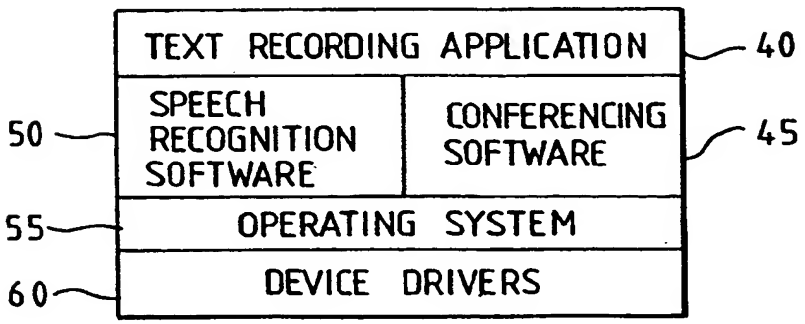

TEXT RECORDING APPLICATION
40
SPEECH RECOGNITION SOFTWARE
CONFERENCING SOFTWARE
45
50
55
OPERATING SYSTEM
60
DEVICE DRIVERS

FIG. 2

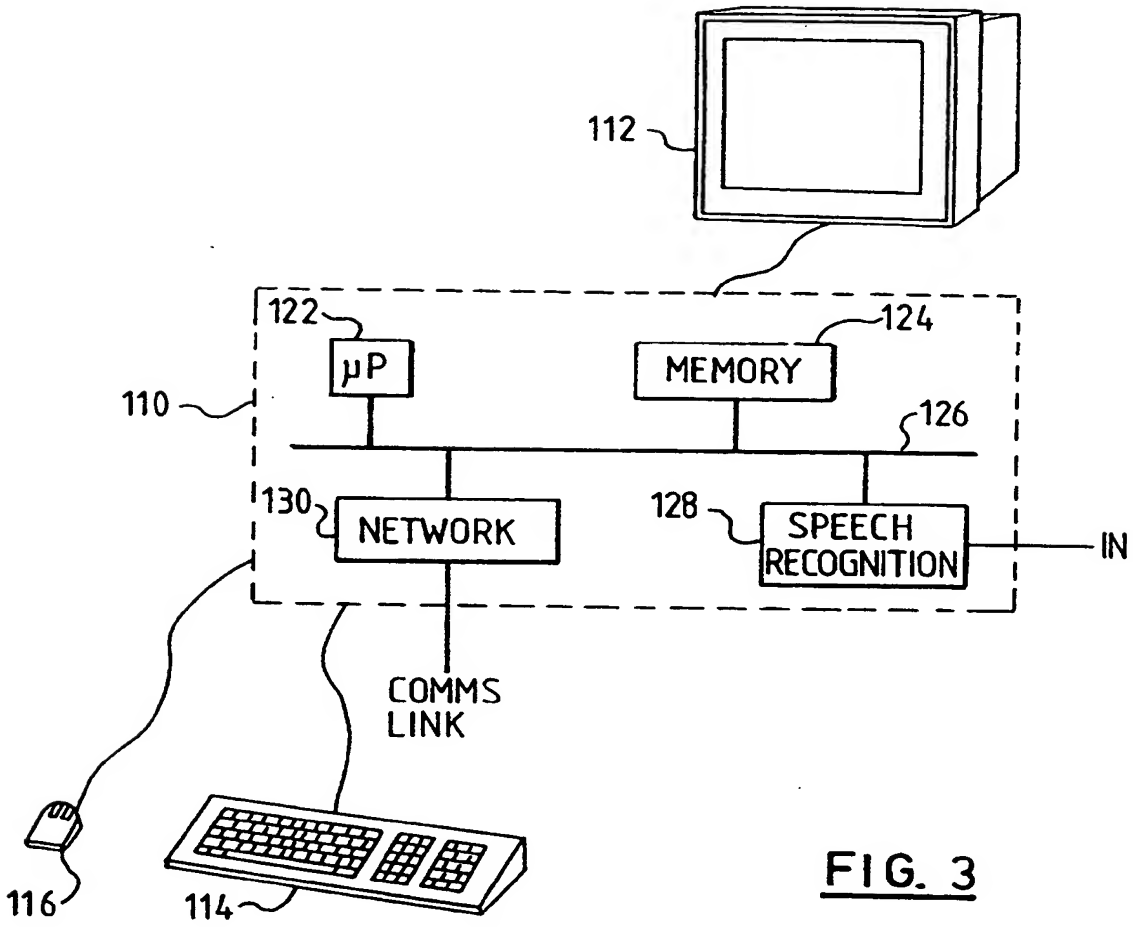

112
122
μP
124
MEMORY
110
126
130
NETWORK
128
SPEECH RECOGNITION
IN
COMMS LINK
116
114

FIG. 3

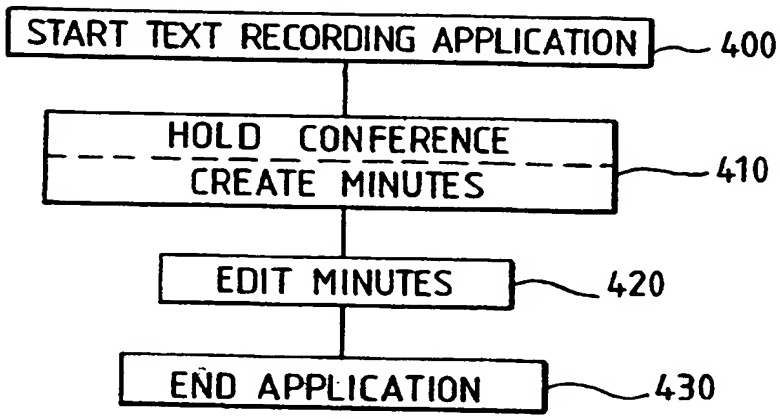

START TEXT RECORDING APPLICATION
400
HOLD CONFERENCE
CREATE MINUTES
410
EDIT MINUTES
420
END APPLICATION
430

FIG. 4

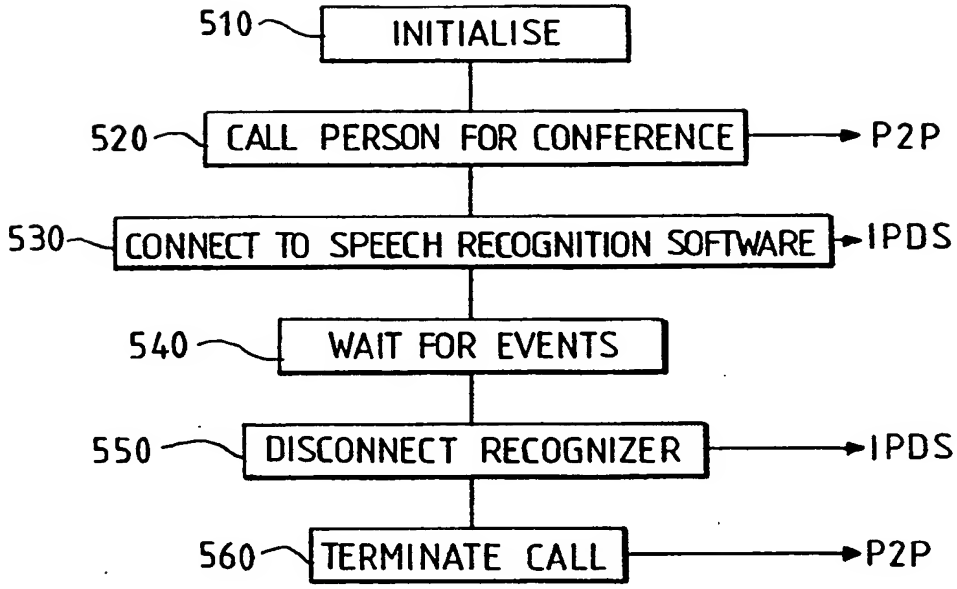


FIG. 5

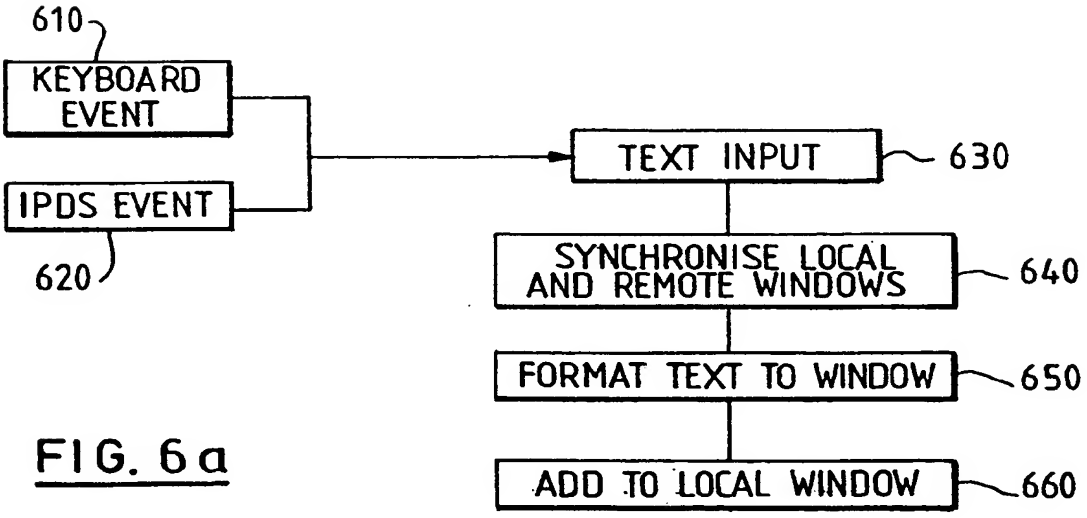


FIG. 6a

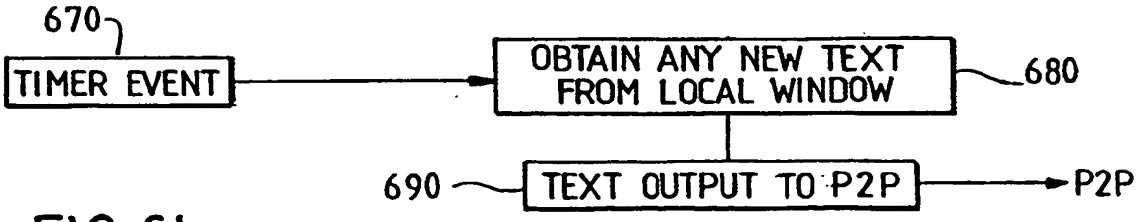


FIG. 6b

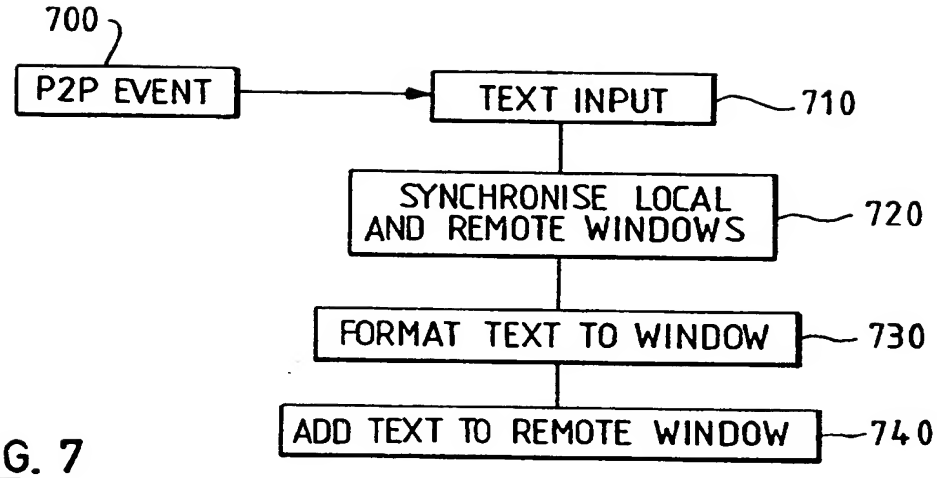

700
P2P EVENT
TEXT INPUT
710
SYNCHRONISE LOCAL AND REMOTE WINDOWS
720
FORMAT TEXT TO WINDOW
730
ADD TEXT TO REMOTE WINDOW
740

FIG. 7

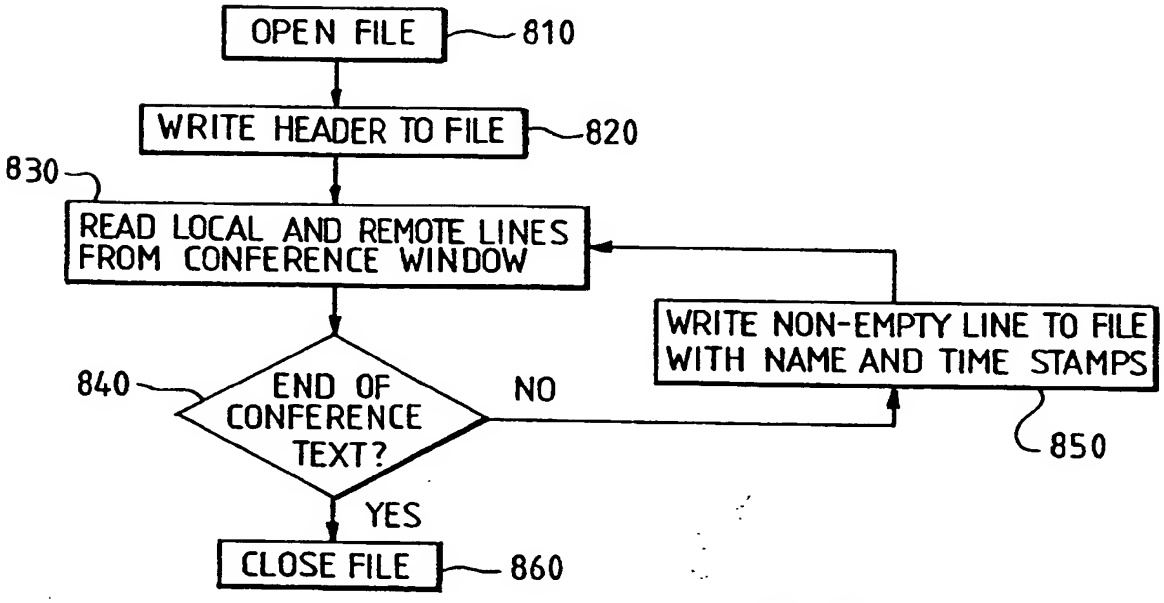

OPEN FILE
810
WRITE HEADER TO FILE
820
830
READ LOCAL AND REMOTE LINES FROM CONFERENCE WINDOW
WRITE NON-EMPTY LINE TO FILE WITH NAME AND TIME STAMPS
840
END OF CONFERENCE TEXT?
NO
850
YES
CLOSE FILE
860

FIG. 8

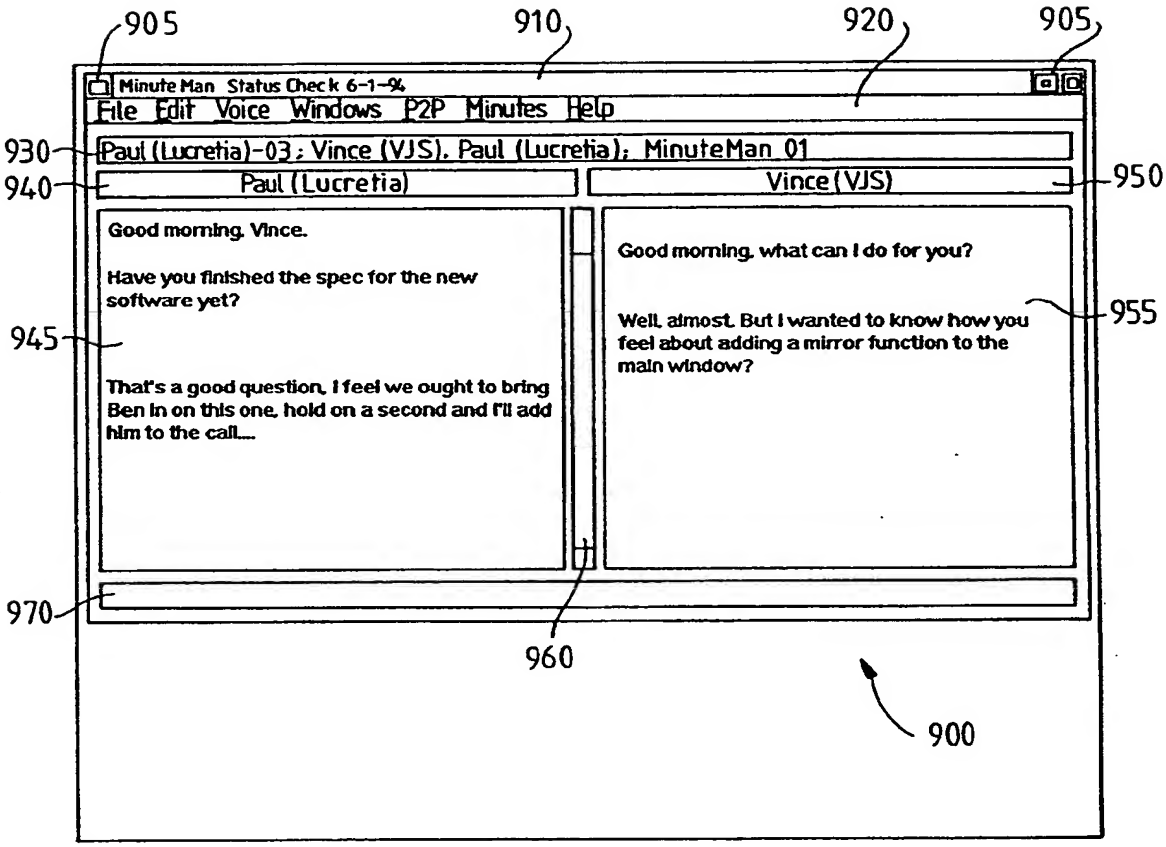

905
910
920
905
Minute Man Status Check 6-1-94
File Edit Voice Windows P2P Minutes Help
930
Paul (Lucretia)-03; Vince (VJS), Paul (Lucretia); MinuteMan 01
940
Paul (Lucretia)
Vince (VJS)
950
Good morning, Vince.
Have you finished the spec for the new software yet?
945
That's a good question, I feel we ought to bring Ben in on this one, hold on a second and I'll add him to the call....
Good morning, what can I do for you?
Well, almost. But I wanted to know how you feel about adding a mirror function to the main window?
955
970
960
900

FIG. 9